VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

GENERÁTOR NELEGITIMNÍHO SÍŤOVÉHO PROVOZU
GENERATOR OF ILLEGITIMATE NETWORK TRAFFIC

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR
Bc. Ondřej Blažek

VEDOUCÍ PRÁCE
SUPERVISOR
Ing. Petr Blažek

BRNO 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**
Ústav telekomunikací

Student: Bc. Ondřej Blažek ***ID:*** 146788
Ročník: 2 ***Akademický rok:*** 2016/17

NÁZEV TÉMATU:

Generátor nelegitimního síťového provozu

POKYNY PRO VYPRACOVÁNÍ:

Diplomová práce je zaměřena na bezpečnost v síťové komunikaci. Cílem práce je vytvořit software, který bude schopen generovat nelegitimní síťový provoz pro testování zranitelnosti sítí. Výstupem práce bude software v jazyce C, který bude generovat vybrané DDoS útoky z transportní až aplikační vrstvy ISO/OSI modelu.

DOPORUČENÁ LITERATURA:

[1] PROWELL, Stacy J., Rob. KRAUS a Mike. BORKIN. Seven deadliest network attacks. Boston: Syngress, c2010. Syngress seven deadliest attacks series. ISBN 15-974-9549-2.

[2] FADYUSHIN, Vyacheslav a Bruce HYSLOP. Instant penetration testing: Setting up a test lab how-to. 1. vyd. Birmingham: Packt Publishing, 2013, 74 s. ISBN 978-1-84969-412-4.

Termín zadání: 1.2.2017 ***Termín odevzdání:*** 24.5.2017

Vedoucí práce: Ing. Petr Blažek
Konzultant:

doc. Ing. Jiří Mišurec, CSc.
předseda oborové rady

ABSTRAKT

Diplomová práce se zabývá problematikou DoS/DDoS útoků a vývojem nástroje, v jazyce C, pro jejich generování. V první kapitole jsou popsány principy DoS útoků mířených na internetovou a trasportní vrstvu ISO/OSI modelu a jsou i podle jejich vlastností rozděleny. Podrobněji jsou zde pak popsány vybrané útoky na aplikační vrstvu a současně s tím také protokoly, na kterých jsou založeny. V nacházející kapitole bylo vytvořeno srovnání volně dostupných nástrojů, které by se dalo použít jako generátory útoků. Praktická část je věnována vývoji nástroje pro generování DoS útoků, zejména pak jeho návrhu, obecnému popisu a ovládání. Dále je v práci uděláno shrnutí nově vzniklé knihovny, včetně výsledků při testování webového serveru, a rozšíření webového rozhraní, které tvoří součást vyvíjeného nástroje.

KLÍČOVÁ SLOVA

DoS, útok, bezpečnost, raw soket, libpcap, vlákno, trafgen, netsniff-ng, ifpps, Nping, packETH, DoSgen, arping, slowloris, sockstress, slow read, záplava

ABSTRACT

The diploma thesis deals with the problems of DoS/DDoS attacks and development of a tool, in C lanugage, for generating them. In the first chapter the principles of DoS attacks targeting the internet and transport layers of ISO/OSI model are described and also according to their characteristics divided. Selected attacks on the application layer are also described here in detail togehter with protocols which they are based on. In the following chapter there has been created a comparison of freely available tools, which could be used as a attack generators. The practical part is dedicated to a development of a tool for DoS attacks, especially design, general description and usage. Further there is a summary of the newly created library, including results of web server testing, and extensions of a web interface, which is part of the developed tool.

KEYWORDS

DoS, attack, security, raw socket, libpcap, thread, trafgen, netsniff-ng, ifpps, Nping, packETH, DoSgen, arping, slowloris, sockstress, slow read, flood

BLAŽEK, Ondřej *Generátor nelegitimního síťového provozu*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2016/2017. 49 s. Vedoucí práce byl Ing. Petr Blažek

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Generátor nelegitimního síťového provozu“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

(podpis autora)

Faculty of Electrical Engineering
and Communication
Brno University of Technology
Purkynova 118, CZ-61200 Brno
Czech Republic

http://www.six.feec.vutbr.cz

PODĚKOVÁNÍ

Výzkum popsaný v této diplomové práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno . . . . . .

. . . . . .
(podpis autora)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce, Ing. Petru Blažkovi, za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno ....

(podpis autora)

OBSAH

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

ÚVOD

Za posledních 20 či 25 let se internet stal natolik součástí našeho života, že se bez něj už lidé nemohou obejít a jeho jakýkoli výpadek nebo omezení může způsobit nepředstavitelné následky a ztráty pro firmy nebo jiné další společnosti. Někdo si pod tím může představit výpadek e-mailové komunikace, nefungováním IP telefonů, ale následky mohou být výraznější, v podobě nefungujících bankomatů nebo online bankovních systémů.

Tyto výpadky mohou být způsobeny nejenom selháním samotných zařízení, či výpadkem elektrického proudu, ale také i cíleným útokem na danou službu. Útoky tohoto typu se nazývají DoS (Denial of Service) či DDoS (Distributed Denial of Service) a jejich cílem je odepřít přístup k počítačovému systému nebo k běžící službě na serveru. V případě těchto útoků nemusí pomoci ani běžné způsoby ochrany systému či počítačových sítí, protože útoky typu DoS jsou často tvořeny tak, aby provoz vypadal co nejlegitimněji. [1, 2, 3]

Tato práce se zabývá bezpečností v síťové komunikaci založené na modelu ISO/OSI, s hlavním cílem vytvořit software v jazyce C, který bude generovat vybrané útoky od transportní po aplikační vrstvu modelu. Vybrány byly útoky HTTP GET flood, Slow Loris, Sockstress a Slow Read.

V první kapitole je rozebrán princip DoS/DDoS útoků z obecného pohledu a následně jsou vysvětleny principy jednotlivých typů útoků na vybranou službu a její charakteristiku.

Druhá kapitola se věnuje popisem a porovnáním nástrojů, z pohledu výkonnosti (rychlost generování paketů), které by mohly být použity jako generátor DoS útoků. Byly vybrány nástroje Trafgen, vyvíjený skupinou netsniff-ng, packETH a nástroj Nping.

Třetí kapitola je věnována návrhu nové knihovny, která by rozšířila stávající nástroj DoSgen převážně o útoky na aplikační vrstvě. Dále je vysvětlen návrh, popis a ovládání vyvíjeného nástroje, spolu se shrnutím a výsledky testování nástroje. Na závěr je provedeno shrnutí informací o nově vzniklé knihovně a vysvětlen princip ovládacího rozhraní.

1 DOS ÚTOKY

Útoky typu DoS (Denial of Service) jsou určené, jak název vypovídá, k odepření služby legitimním uživatelům (klientům). Jejich hlavním cílem je zahltit cílovou síť, nějakou službu, či jinou webovou aplikaci určitým množstvím cíleně podvržených požadavků, že tato oběť nebude schopna odpovídat na požadavky oprávněné. Každý útok je něčím charakteristický a nelze tedy všechny zařadit do stejné kategorie.[1]

1.1 Typy DoS útoků

Většina DoS útoků je založená buďto na protokolu TCP nebo UDP, mířící na transportní nebo aplikační vrstvu TCP/IP modelu.
Prvním možným rozdělením DoS útoků pak může být na:

- Spojově orientované: Útok, který nastává v případě, že bylo vytvořeno spojení mezi klientem a serverem s pomocí některého ze spojových protokolů.
- Bezespojové: Tento typ nevyžaduje sestavení spojení mezi klientem a serverem, předtím než je možné vykonat útok.

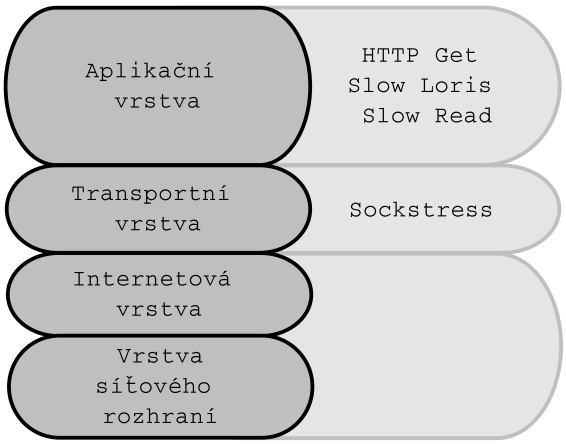
Je mnoho způsobů, jak útočit na cílový server. Může se jednat o zneužití nedostatků internetových protokolů nebo zahlcení oběti nelegitimními požadavky, čímž je zabráněno aby server dostal požadavky oprávněné. V praxi je však nejvíce rozšířeno dělení se zaměřením na:

- Síťové zdroje (Záplavové/Pomalé)
- Serverové zdroje (Slabina v TCP/IP)
- Aplikační zdroje (Sofistikované útoky na cílovou službu nebo aplikaci, využívající nějakou jeji slabinu) [1, 2]

Tato práce se zaměřuje převážně na útoky posledního typu, tedy útoky na aplikační vrstvu, jelikož jsou nejhůře detekovatelné a v poslední době velice rozšířené. Konkrétně se jedná o útoky cílené na HTTP protokol, kromě útoku typu Sockstress, který je cílený na transportní vrstvu TCP/IP modelu, jak lze vidět na obr.1.1.
Seznam vybraných útoků je následující:

- Sockstress
- HTTP GET flood
- Slow Loris
- Slow Read

V následujících podkapitolách budou vysvětleny principy těchto útoků, případně protokoly na které jsou tyto útoky mířené.



Obr. 1.1: Rozdělení útoků podle TCP/IP modelu

1.2 Útoky mířené na síťové zdroje

Cílem tohoto útoku je zaplavit cílovou síť takovým množstvím provozu, až je spotřebována většina šířky pásma dané sítě. Útoky tohoto typu jsou nazývány „Záplavové". V praxi je jich většinou docíleno s použitím botnetu, tzv. „DDoS útokem", armádou počítačů ovládanou útočníkem, viz obr.1.2. Útočník distribuuje nástroj mezi své oběti ve formě malwaru a ty pak, aniž by o tom věděly, se stanou jedním ze zdrojů útoku.

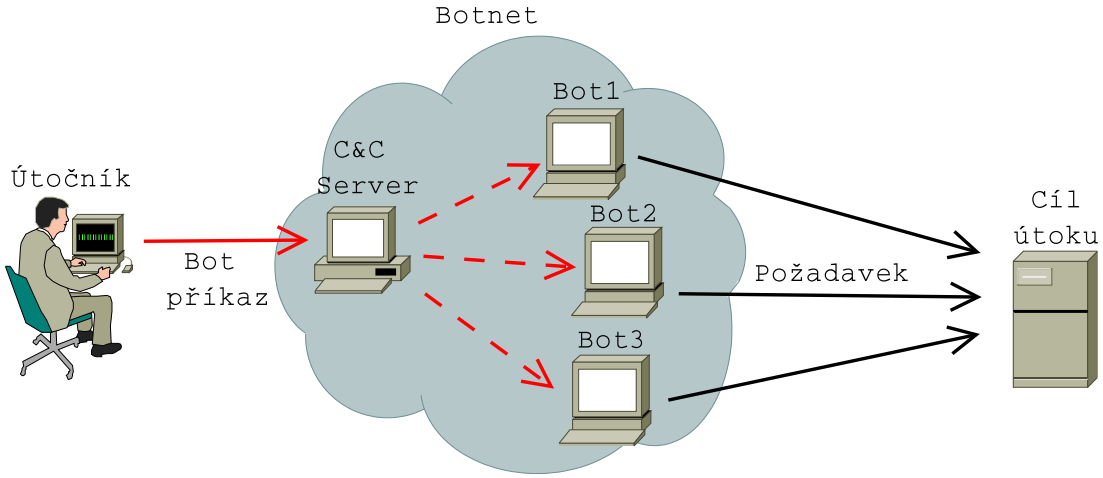
1.2.1 Záplavové

Jedním z nejběžnějších je útok využívající protokol UDP, bezestavový protokol využívající protokolu IP, nižší vrstvy, k zapouzdření svých datagramů. Útok je nazýván „**UDP Flood**" a jeho princip spočívá v jednoduchém přenosu dat mezi dvěma stranami, bez nutnosti sestavit spojení[1].

Útočník odesílá velké množství UDP datagramů, s náhodně zvolenými IP adresami a náhodným cílovým portem. Server pak musí zpracovat každý z datagramů a odpovídá na ně zprávou ICMP typu „Cíl nedosažitelný", čímž se spotřebuje velké množství šířky pásma.

Druhým z těch běžných záplavových útoků je útok „**ICMP/Ping flood**", který funguje na podobném principu, jen s tím, že odesílá nějakou ICMP zprávu, např. „Echo request", a jakmile server obdrží spoustu takovýchto zpráv, tak podobně jako u předchozího útoku musí taky zpracovat každou ze zpráv a tím dochází k DoS útoku. [3]

[1] Jako je tomu např. u služeb využívajících protokol TCP.



Obr. 1.2: DDoS útok

1.3 Útoky mířené na serverové zdroje

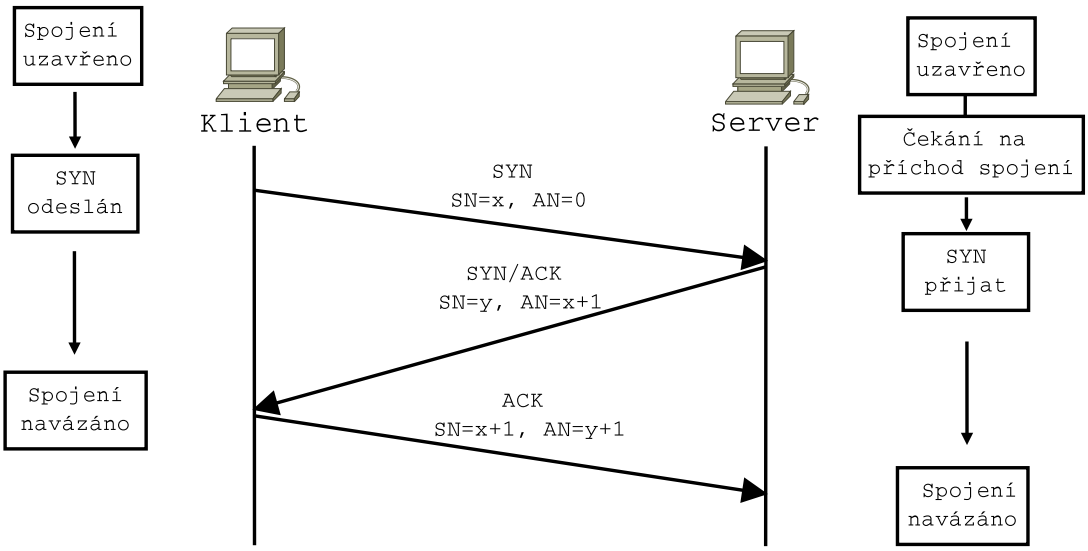
Jejich cílem je vyčerpat výpočetní kapacity serveru a způsobit tak DoS útok. Princip je postaven na tom, že útočník využije nějaké slabiny protokolu využívaného k běžné komunikaci (TCP) a zahltí server různými požadavky, kterými se server bude nucen zabývat až do doby, kdy mu dojdou výpočetní kapacity a nebude už schopen reagovat na požadavky od oprávněných uživatelů. [1]

1.3.1 Běžná TCP komunikace

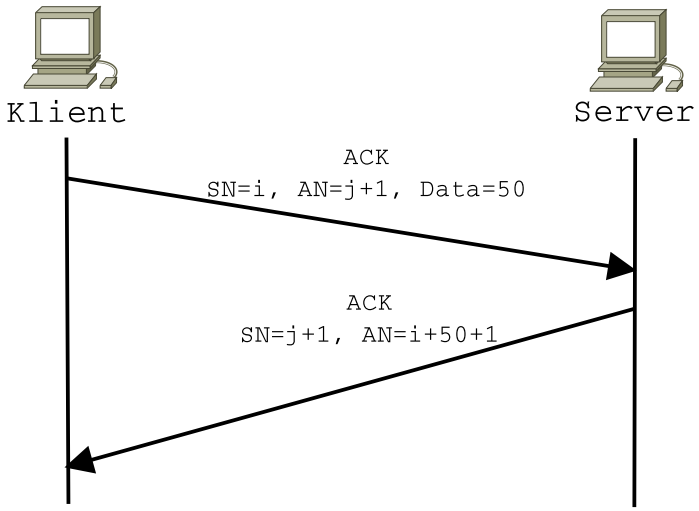
Většina DoS útoků je v této práci zaměřena na některý z aplikačních protokolů, které jsou založené na TCP protokolu nebo přímo útočí na nějakou jeho slabinu, je nutné pro úplnost zmínit jak probíhá běžná TCP komunikace.

Protokol TCP je založen na spolehlivém přenosu dat a zároveň zajišťuje bezchybnost či doručení ve správném pořadí. Názorná ukázka komunikace je možná vidět na obr. 1.3 a 1.4. Klient inicializuje spojení zasláním segmentu s příznakem SYN, nastaveným na binární jedničku. Mimo to odešle sekvenční číslo SN, díky kterému je možné zajistit správné seřazení a úplnost přijatých dat. Potvrzovací číslo AN je při navazování spojení nastaveno na nulu, v opačném případě bude obsahovat SN číslo předešlého TCP segmentu daného spojení, zvýšeného o jedničku.

V případě zasílání dat je číslo SN zvýšeno o jedničku, dle předchozího segmentu daného spojení. Číslo AN je zvýšeno o jedničku, plus počet bytů přijatých dat předchozího segmentu. Dalším z bloků TCP záhlaví je délka okna W, kterou klient specifikuje serveru maximální počet bajtů, které je schopen zpracovat. Jak je uvedeno v jedné z dalších kapitol 1.3.3, tohoto bloku může být zneužito pro nelegitimní účely. [4, 5]



Obr. 1.3: Navázání TCP komunikace



Obr. 1.4: Odesílání dat pomocí protokolu TCP

1.3.2 Využívající slabinu TCP/IP

Běžně je využíváno slabin v protokolu TCP/IP, který nemá žádnou ochranu např. proti MITM útokům. Útočník je např. schopen zrušit sestavené spojení nebo vytvářet fiktivní spojení se serverem tak, aby vyčerpal jeho prostředky. Používá k tomu příznakové bity (SYN, ACK, RST, PSH, FIN nebo URG), které jsou specifikované uvnitř TCP segmentu.

Hlavním představitelem je tzv. „**TCP SYN flood**“ útok. Předtím než spolu dvě strany pomocí TCP protokolu mohou komunikovat je nutné vytvořit tzv. „three way handshake“, neboli třícestné podání ruky, a až poté je možné odesílat aplikační data.

Útok jednoduše využívá toho, že server očekává legitimní navázání spojení pomocí příznaku SYN, uvnitř TCP segmentu, a ke každému takovému požadavku vyčlení prostor v paměti, včetně vytvoření nového vlákna, které by toto spojení obsloužilo. Server má omezené kapacity a v případě masivního počtu požadavků o vytvoření spojení dojde k vyčerpání této kapacity a úspěšnému DoS útoku. Je to z důvodu toho, že server při otevření spojení odešle odpověď SYN+ACK a pak čeká určitou dobu na příchod třetí zprávy ACK na sestavení spojení.

Jinou možností je využití příznaků **PSH+ACK**, kdy útočník nastaví ve vysílaných segmentech PSH bit na 1 a daný segment je pak okamžitě poslán do TCP zásobníku na serveru, který je ale těsně předtím vyprázdněn. Jakmile je tento krok hotov, server posílá ACK zprávu nazpět. Stějně jako v případě „TCP SYN flood“ je tento útok v případě velkého počtu příchozích segmentů úspěšný. [3]

1.3.3 Pomalé útoky typu Sockstress

Na rozdíl od záplavových neposílají takové množství zátěže na servery, ale vypadají jako běžný provoz, které jsou však mířené na specifickou slabinu nějaké služby.

Jedním takovým je útok na TCP zásobník, který rovněž způsobuje odepření služby. V normální TCP komunikaci je nutné navázat spojení, tzv. trojcestné „podání ruky“, předtím než je možné odesílat data serveru. Je tedy odeslán segment s příznakem SYN, pro navázání spojení, a server odpovídá segmentem SYN+ACK. První strana pokračuje pak už jen segmentem s příznakem ACK a následuje odesílání dat. V tomto útoku je v uvnitř posledního segmentu (ACK) nastavena „Velikost okna“[2] na hodnotu 0, čímž je řečeno kolik místa klientu zbývá ve své vyrovnávací paměti pro uložení dat. Server pak je nucen zastavit odesílání dat, dokud mu druhá strana neřekne jinak a odesílá jen pravidelně klientovi zprávu s otázkou, kdy dostane nové informace . Útočník však nemá v plánu nic měnit a server tak čeká donekonečna. V případě spousty takovýchto spojení je dosaženo DoS útoku a vyčerpání zdrojů serveru. [6]

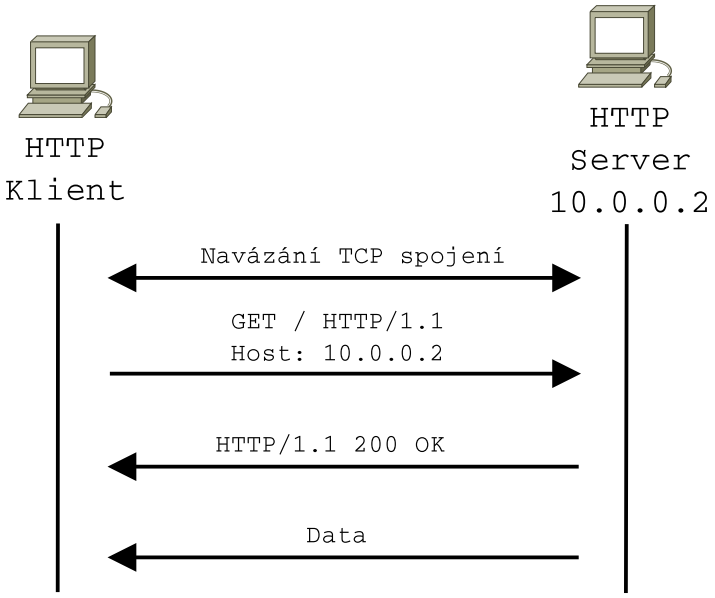
1.4 Útoky mířené na aplikační zdroje

Útoky tohoto typu v poslední době vzrostly a lze se tak setkat z útoky na nejeden aplikační protokol. Může se jednat o HTTP, HTTPs, DNS, SMTP a další, s tím, že útoky nevyužívají jen záplavovou techniku 1.2.1, ale i pomalou 1.3.3. [1, 3]

[2]Součást TCP segmentu.

1.4.1 Protokol HTTP

Hlavním protokolem na který jsou cílené DoS útoky je HTTP, je proto nutné zde zmínit na jakém principu je založen. Pro svoji potřebu používá protokol TCP, z transportní vrstvy, na portu 80. Používá na přenos dokumentů ve formátu HTML, či jiného libovolného typu dat. O data je žádáno z klientské stanice pomocí tzv. „dotazů“, na které server odpovídá statusovými kódy, včetně stavového hlášení a případně odpovídajícími daty.[3] Komunikace je znázorněna na obrázku 1.5. [4, 7]



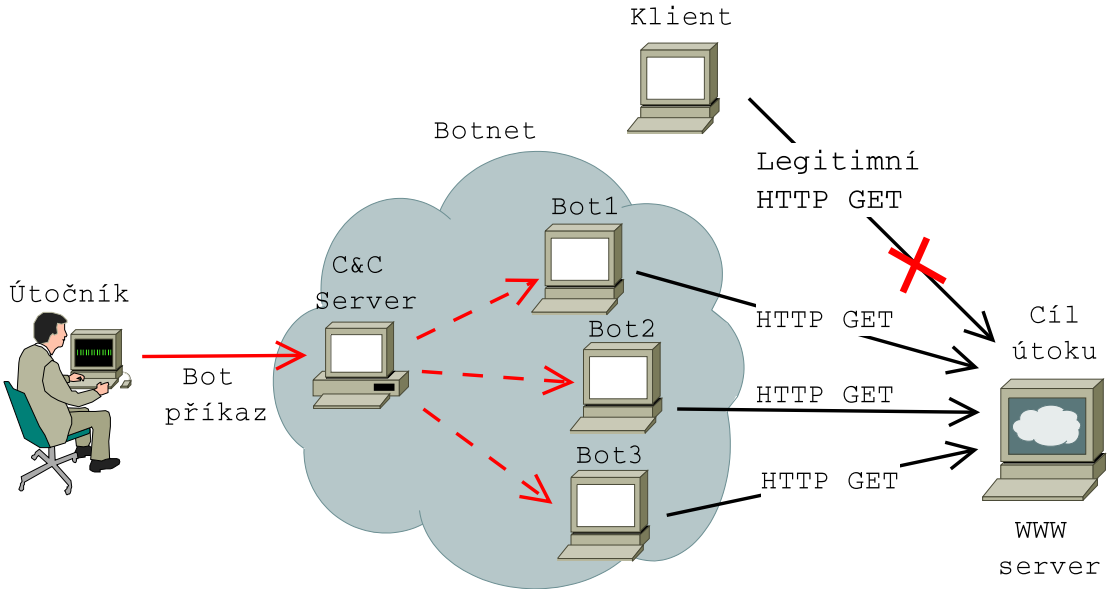
Obr. 1.5: HTTP komunikace

1.4.2 HTTP Flood

Tento útok se skládá z obyčejných HTTP GET požadavků, které se na první pohled jeví jako legitimní, což je dělá obtížnými detekovat. Je obvykle vykonáván z mnoha stanic, které současně odešlou požadavek o stažení kořenové stránky, či nějakého specifického obsahu. Tím, že cílený soubor může být velký, server ho pak musí načíst z pevného disku do paměti a postupně, třeba i v několika paketech, ho odeslat klientovi. To může způsobit celkem výrazný vliv na zátěž disku, CPU, paměť či samotnou síť a na oprávněné HTTP požadavky pak nemusí zbýt v paměti místo. Princip útoku je možné vidět na obrázku 1.6. [2, 3]

Samotná detekce a příp. blokování se sice může jevit jako jednoduché, ale pokud je použita kombinace požadavků na různý obsah stránky, tak je detekce podstatně stížena.

[3]Seznam možných statusových kódů a jejich významů je uveden v `https://tools.ietf.org/html/rfc7231#page-49`



Obr. 1.6: HTTP GET Flood

1.4.3 Slow Loris

Útok typu Slow Loris je jedním z tzv. pomalých útoků 1.3.3, avšak mířených na aplikační vrstvu, vytvořen hackerem se jménem „RSnake“. Tento útok, potažmo nástroj, spočívá v principu odeslaní částečných HTTP požadavků tak, aby nedošlo k vypršení TCP spojení, ale aby webový server dostal další část HTTP zprávy těsně před vypršením daného spojení. Je tedy možné odeslat jen část HTTP požadavku, např. jen první řádek dotazu (GET / HTTP/1.1) a v následujících krocích další části. Server je tak nucen čekat neustále na další části zprávy, čímž se na libovolně dlouho dobu, udávanou v tomto případě útočníkem, stane nedostupným. V případě, že je pak vytvořen dostatek spojení, dochází k DoS/DDoS útoku. [3, 8]

1.4.4 Slow Read

V případě útoku Slow Read klient naváže TCP spojení obvyklým způsobem, avšak uvnitř HTTP požadavku oznámí velikost okna[4], pro příjem dat, na nějakou malou hodnotu. Server je pak nucen data rozdělit na několik malých dílů, v závislosti na velikosti požadovaných dat, a tyto data pak pomalu odesílat klientovi. Ten každou ze zpráv potvrdí příznakem ACK a oznámí serveru opět velice malé číslo uvnitř bloku „velikost okna“, čímž nutí server data odesílat stále stejně pomalu. Je tak možné odeslat požadavek na nějaký velký soubor, kdy bude server v případě většího počtu spojení zaneprázdněn zpracováním požadavků na neobvykle dlouhou dobu. [3, 9]

[4]Vysvětleno v podkapitole 1.3.1

2 NÁSTROJE PRO DOS ÚTOKY

Nástrojů pro provedení DoS/DDoS útoků je dostupných mnoho, byly z nich tedy vybrány ty, které splňují požadavek být napsané v jazyce C a mít otevřený zdrojový kód. Vybrané z nich zde budou představeny a porovnány v zátěžovém testu.

2.1 PackETH

Jedním z možných volně dostupných nástrojů pro generování paketů na Ethernetu je packETH. Jedná se o bezestavový nástroj, vytvořený v jazyce C, který umožňuje odesílat pakety ven ze zařízení a nezáleží mu na případné odpovědi od protějšku. Umožňuje odesílat posloupnost paketů a během odesílání měnit parametry, jako jsou IP adresy, MAC adresy aj. Mimo jiné podporuje ukládání konfigurace do souboru a načítání z něho. [10]

```
$ sudo apt-get install gcc pkg-config libgtk2.0-dev libglib2.0-
    dev make
```

Po stažení závislostí je možné nástroj rozbalit a pomocí nástroje automake zkompilovat.

```
$ tar xvf packETH-1.8.1.tar.bz2
$ cd packETH-1.8.1/
$ make
```

Následně je možné pomocí uživatelsky přívětivého rozhraní vytvořit vlastní paket a odeslat jej ven ze zařízení, přes specifikovaný port. Provádí se to pomocí následujícího příkazu:

```
$ sudo ./packETHcli -i eth0 -m 2 -d 0 -n 200 -f packet1.pcap
```

Argument `-i` určuje port, přes který jsou pakety odeslány. Druhý argument `-m`, s hodnotou `2`, značí mód programu (posloupnost), `-d` je pro zpoždení v µs, `-n` definuje počet paketů k odeslání a `-f` načte soubor, ve formátu .pcap, z kořenové složky.

2.2 Trafgen

Trafgen je volně dostupný nástroj, který je součástí balíčku Netsniff-NG. Jedná se o velice rychlý, vícevláknový, nízkoúrovňový nástroj pro generování paketů, využívající tzv. „zero-copy“ mechanismus. To usnadní kopírování paketů tak, že není nutné je kopírovat z jádra operačního systému do uživatelského prostoru a opačně. Značně se tím omezí čas a prostředky nutné pro odesílání a přijímání paketů. Trafgen není limitován žádným protokolem, díky jeho konfiguračnímu jazyku je možné ho použít

jakkoli. [11, 12]
Ke zkompilování programu jsou nutné následující závislosti:

```
$ sudo apt-get install git flex bison libnl-3-dev libnl-genl-3-
    dev pkg-config
```

Po stažení závislostí je možné stáhnout a zkompilovat balíček nástrojů netniff-ng:

```
$ git clone https://github.com/borkmann/netsniff-ng
$ cd netsniff-ng/
$ make trafgen
```

Trafgen má svůj vlastní, tzv. „low-level“ konfigurační jazyk pro vytváření paketů založený na makrech, vytvořených v jazyce C. Umožňuje ale také vytvořit pakety ze zachycených pcap souborů. Díky jeho konfiguračnímu jazyku je možné sestavit vlastní paket s libovolnou délkou a obsahem.
Příklady maker jsou uvedeny v následující tabulce:

Tab. 2.1: Příklady maker nástroje trafgen

Definice makra	Význam
fill(m, n)	Naplň n bytů hodnotami m.
rnd(n)	Vygeneruj náhodně n bytů
csumip(n, m)	Vypočítej kontrolní součet pro IP hlavičku
constm(n)	Ulož m bitovou konstantní hodnotu n, n = {8, 16, 32, 64}

Vytvořený paket (soubor s příponou .cfg nebo .txt) je možné odeslat např. pomocí následujícího příkazu:

```
$ sudo ./trafgen --in packet1.cfg --out eth0 --num 1000
```

Argument `--num` značí počet paketů k odeslání a argument `--out` port, přes který jsou pakety odeslány ven.

2.3 Nping

Nping je volně dostupný síťový nástroj pro generování paketů typu Ethernet a pro analýzu chování sítě. Podporuje protokoly TCP, UDP, ICMP i ARP, včetně experimentální podpory IPv6. Je běžně použit pro testování dostupnosti cílových stanic, pomocí protokolu ICMP, ale zároveň umožňuje vytvářet tzv. „raw“ pakety pro výkonové testování sítě, či DoS útoky. Nástroj má velice flexibilní rozhraní přes příkazovou řádku, ale je možné zvolit i grafickou nadstavbu. Nping je součást většího nástroje Nmap (Network Mapper), který je běžně použit pro objevování zařízení na síti nebo bezpečnostní audity. [13]

K samotné kompilaci programu jsou zapotřebí standartní knihovny jazyka C, včetně překladače gcc a automatizačního nástroje make:

```
$ sudo apt-get install gcc g++ make
```

Nping lze získat stažením následujícího archivu a kompilací zdrojových souborů:

```
$ wget https://nmap.org/dist/nmap-7.31.tar.bz2
$ bzip2 -cd nmap-7.31.tar.bz2 | tar xvf -
$ cd nmap-7.31/
$ ./configure
$ make
$ sudo make install
```

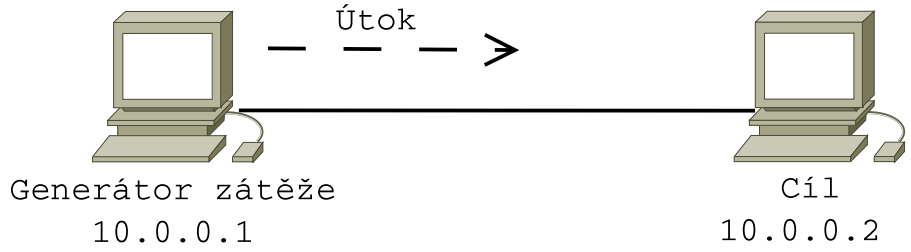
Spuštění programu je provedeno příkazem:

```
$ sudo ./nping --icmp --icmp-type 8 10.0.0.2 --rate 999999 -H -N
    -e eth0 -c 3000000
```

Argument `-H` skryje výpis odeslaných paketů a argument `-N` zamezí pokusy o zachytávání odpovědí.

2.4 Výkonové testování nástrojů

Jednotlivé nástroje byly porovnány takovým způsobem, aby bylo možné zjistit, který z nich dokáže nejefektivněji generovat zátěž. Měření bylo prováděno jak straně zdroje, který generuje zátěž, tak na straně cíle, viz obr.2.1, aby byly ošetřeny nežádoucí odchylky v měření. Pro srovnání bylo měřeno na Fast Ethernetu i na Gigabit Ethernetu, v jednotkách pps (paket za sekundu), včetně vytížení linky v MB/s.



Obr. 2.1: Zapojení přístrojů pro zátěžové testování

Pro potřeby testování byl použit síťový statistický nástroj ifpps, který získává statistiky přímo ze souborů procfs, z linuxového jádra. Díky tomu tak nedochází, při vysokých rychlostech, ke zkreslení výsledků, jako je tomu v případě programů, které spadají pod prostor uživatele (např. použití knihovny libpcap). Ifpps je obsažen v balíčku netsniff-ng, ale je možné jej zkompilovat i samostatně.

Měření bylo soustředěno na generování provozu s použitím protokolů TCP, UDP a ICMP. Verze nástrojů byly následovné:

- packETH - 1.8.1
- Trafgen - 0.6.2+
- Nping - 0.7.01

Jako generátor zátěže bylo použito PC s následujícími specifikacemi:

- procesor: Intel Core i5-4300U CPU @ 1.90GHz, 3MB cache, 2 jádra,
- RAM: 8 GB,
- OS: Ubuntu 16.04.1 LTS, 64bit, kernel: 4.4.0-45-generic

2.4.1 Výsledky měření

Naměřené hodnoty, při použití všech tří nástrojů, byly uvedeny v samostatných tabulkách. Aby byly měřené hodnoty co nejméně ovlivněny, byly odečteny tři hodnoty pro každé měření a udělán jejich průměr. Až tyto hodnoty pak byly zaneseny do tabulek a grafů.Na první z tabulek tab. 2.2 lze vidět, že nástroje packETH a Trafgen byly srovnatelně výkonné a počet vygerenovaných paketů za sekundu se pohyboval pro 1 GbE kolem 380 000. V druhé tabulce 2.3 lze vidět nárůst výkonu v případě nástrojů Trafgen a packETH přes 20%, při měření protokolu TCP. Pro poslední z testovaných protokolů UDP, uvedeným v tab. 2.4, byl naměřen rozdíl cca 170 000 paketů za sekundu, ve prospěch Trafgenu, ve srovnání s ostatními nástroji.

Tab. 2.2: Naměřené hodnoty pro protokol ICMP

Nástroj	Počet paketů/sek [pps]		Vytížení linky [MB/s]	
	Fast Ethernet	Gigabit Ethernet	Fast Ethernet	Gigabit Ethernet
packETH	141 924	372 301	8,66	23,21
Trafgen	141 913	386 356	8,65	23,45
Nping	141 823	145 080	8,64	8,05

Tab. 2.3: Naměřené hodnoty pro protokol TCP

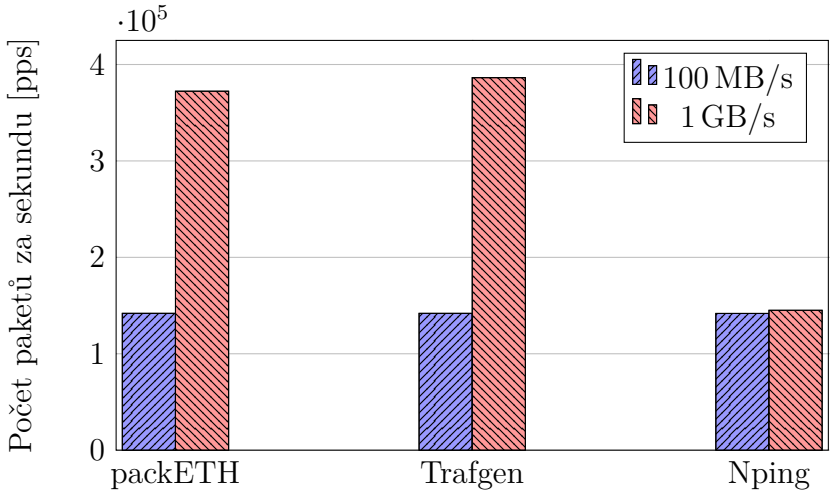
Nástroj	Počet paketů/sek [pps]		Vytížení linky [MB/s]	
	Fast Ethernet	Gigabit Ethernet	Fast Ethernet	Gigabit Ethernet
packETH	148 589	800 401	8,81	45,75
Trafgen	148 618	800 572	8,81	45,75
Nping	141 831	159 610	8,63	9,75

Tab. 2.4: Naměřené hodnoty pro protokol UDP

Nástroj	Počet paketů/sek [pps]		Vytížení linky [MB/s]	
	Fast Ethernet	Gigabit Ethernet	Fast Ethernet	Gigabit Ethernet
packETH	148 614	952 241	8,82	54,64
Trafgen	148 612	1 127 542	8,81	64,72
Nping	141 831	167 242	8,63	11,3

Při nastavování parametrů sítě bylo nutné ručně nastavit rychlost sítové karty, konkrétně při testování přenosu přes 100 Mb Ethernet. Síťový nástroj Ethtool umožňuje měnit parametry pro síťovou kartu nebo síťový ovladač:

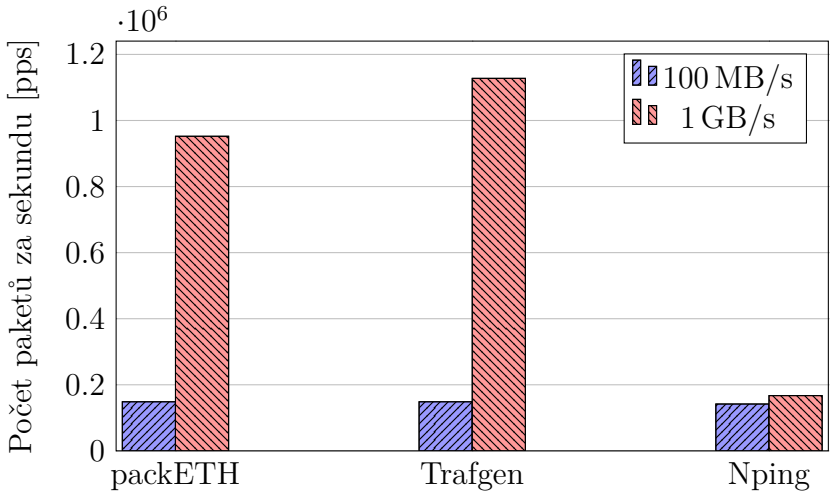
```
$ sudo ethtool -s eth0 speed 100 advertise 0x008 duplex full
    autoneg on
```



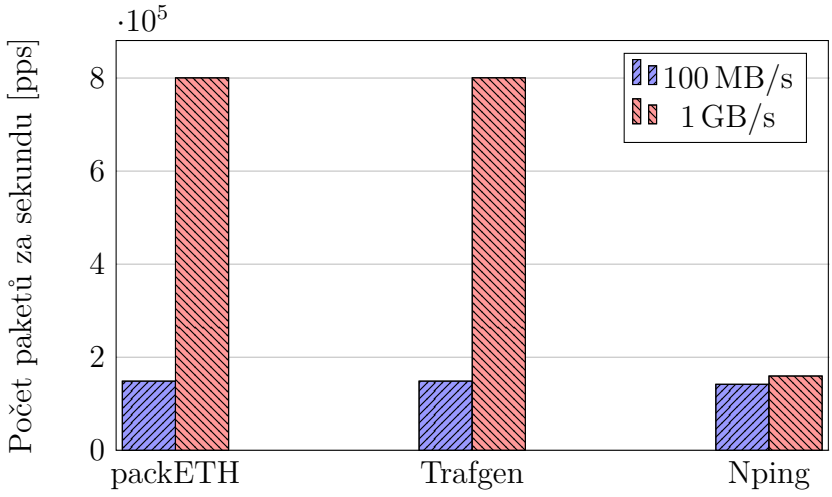
Obr. 2.2: Porovnání nástrojů z hlediska přenesených paketů pro protokol ICMP

Po vyhodnocení měření je možné říci, že nejlepší výsledek (největší počet odeslaných paketů za sekundu) byl dosažen nástrojem Trafgen. Jak je možné vidět v grafech 2.2, 2.3 a 2.4, pro 100 MB/s nebyl rozdíl téměř žádný a všechny nástroje dokázaly plně využít šířku pásma. V případě nástroje Trafgen tedy nezáleželo na tom kolik vláken je využito, protože ostatní nástroje dokázaly, i s použitím jednoho, vygenerovat stejné množství paketů. Větší rozdíly pak nastaly při měření 1 GB/s, kdy rozdíly dosáhly až 85%, konkrétně to bylo mezi nástrojem Nping a Trafgen. Rozdíly mezi nástrojem packETH a Trafgen nebyly natolik výrazné, což může být způsobeno nedostatečným výkonem PC, které bylo použito jako generátor provozu.

Za hlavní příčinu úspěchu nástroje Trafgen se dá považovat podpora více jader nebo použité zero-copy mechanismy.



Obr. 2.3: Porovnání nástrojů z hlediska přenesených paketů pro protokol UDP



Obr. 2.4: Porovnání nástrojů z hlediska přenesených paketů pro protokol TCP

3 VÝVOJ NÁSTROJE PRO DOS ÚTOKY

Dalším bodem práce bylo vytvořit software pro generování nelegitimního provozu v jazyce C, který by byl schopen generovat vybrané DoS/DDoS útoky z transportní až aplikační vrstvy ISO/OSI modelu. Pro tyto účely byl zvolen nástroj Trafgen, jako jádro vlastního generátoru, jelikož měl nejlepší výsledky, jak uvádí kapitola 2.4, a je také nejflexibilnější z vybraných tří.

Jak bylo uvedeno v sekci 2.2, Trafgen načítá konfiguraci paketů ze souboru, je založen na vlastním jazyce výrazů a makrech jazyka C. Hlavičky paketů i samotná data je možné vytvořit v běžném editoru a definovat jej pomocí maker mezi dvěma složenými závorkami, oddělenými čárkou.

```
1 {
2 ...definice paketu1...
3 }
4
5 {
6 ...definice paketu2...
7 }
```

Umožňuje také specifikovat, který procesor použít, při běhu programu, volbou `--cpus` při spouštění programu. Samotný konfigurační jazyk umožňuje definovat hodnoty v soustavě binární, hexadecimální, oktalové i dekadické, ale také má podporu pro řetězce a znaky (viz tab. 3.1). Trafgen má rovněž nadefinovaná makra pro

Tab. 3.1: Příklady datových typů v nástroji Trafgen [14]

Datové typy	Příklad
Binární	`0b0001, 0b10001001`
Hexadecimální	`0x00, 0xff`
Oktalové	`0213, 063247`
Dekadické	`8, 8888`
Řetězce	`"retezec", "TRAFGEN"`
Znaky	`'n','0'`

vygenerování náhodných dat pro každý odeslaný paket, díky tomu je možné např. změnit zdrojovou MAC adresu u každého odeslaného paketu (viz tab. 3.2).

Díky tomu, že Trafgen posílá konfigurační soubory přes preprocesor jazyka C, je možné nadefinovat pomocí příkazu `#define` makra, např. makra v hlavičkovém souboru `trafgen_stddef.h`, které jsou po zpracování nahrazeny danou hodnotou. Ve výsledku je pak uvnitř konfiguračního souboru nahrazeno makro `TCP_FLAG_SYN` hodnotou `0000 0010`[14].

Tab. 3.2: Příklady maker umožňujících změnu dat v průběhu programu[14]

Definice makra	Popis
drnd(n)	Vygeneruj náhodná data při běhu programu
dinc(m, n, o)	Inkrementuj hodnotu `m` o `n`, pro `o`-bytové pole při běhu programu
ddec(m, n, o)	Dekrementuj hodnotu `m` o `n`, pro `o`-bytové pole při běhu programu

Bylo rozhodnuto rozšířit stávající nástroj DoSgen, který vzešel z myšlenky použít Trafgen jako jádro generátoru a vytvořit tak generátor DoS útoků.

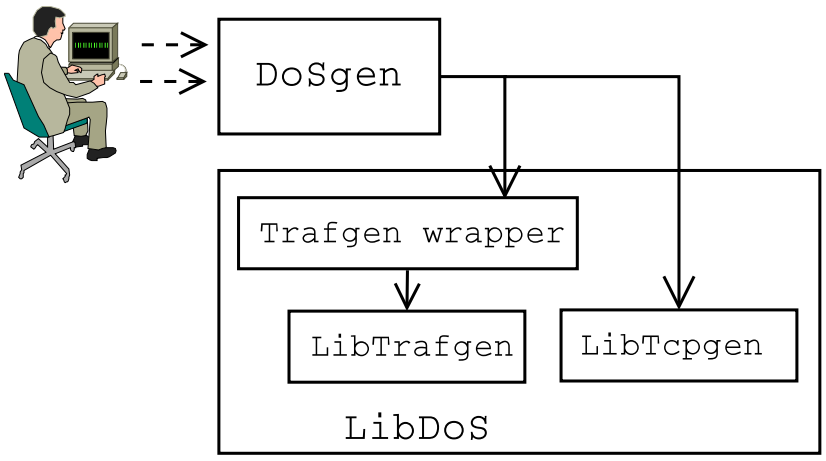
3.1 Struktura

Nástroj DoSgen byl původně složen ze dvou hlavních modulů:

- `DoSgen` - hlavní rozhraní mezi uživatelem a jádrem programu,
- `LibDoS` - knihovna, obsahující moduly `Trafgen wrapper` a `LibTrafgen`.

Tím, že při vývoji nástroje Trafgen nebylo počítáno s tím, že by mohl být použit pro sestavování TCP spojení, tak tento bod nebyl nijak implementován. Trafgen je a byl využíván čistě jako rychlý generátor paketů.

Jelikož byla tato práce zaměřena převážně na rozšíření nástroje o útoky na aplikační vrstvě, bylo tak nutné zmíněný bod, o sestavení TCP spojení, nějakým způsobem implementovat. Pro tyto účely byl v této práci vytvořen nový multivláknový modul `Tcpgen`, vytvořen jako statická knihovna, který by obsahoval nutné funkce pro sestavení a případného udržování TCP spojení. Modul `LibDoS` tak byl rozšířen o tuto statickou knihovnu. Nově rozšířený nástroj pak obsahuje dvě jádra (`trafgen` a `tcpgen`). Jeho strukturu je možné vidět na obr. 3.1.



Obr. 3.1: Struktura nástroje DoSgen[15]

3.1.1 DoSgen

Jedná se o hlavní soubor programu, jakož i název, který má na starost zpracování vstupních parametrů a samotnou interakci s uživatelem. Především se jedná o základní chybové ošetření, výpisy na obrazovku, formátování vstupních proměných a volání následujícího modulu, v závislosti na zvoleném útoku.

3.1.2 Trafgen wrapper

Uvnitř modulu dochází ke zpracování volitelných či povinných parametrů a jejich zápis, společně s souborem `trafgen_configs`, do dočasného konfiguračního souboru. Ten představuje šablonu, podle které je v závislosti na zvoleném útoku vytvořen paket. Dochází zde také i k ověřování správného otevření, zápisu a uzavření konfiguračního souboru. Dočasný soubor je pak jako parametr předán hlavní funkci nástroje Trafgen, který začne s generováním specifikovaných paketů.

3.2 Tcpgen

`Tcpgen` byl navržen tak, aby bylo možné vytvářet vlastní hlavičky IP a TCP vrstev, což znamenalo použít raw sokety. To umožní vystupovat z útočícího stroje s fiktivní IP adresou, něž je přidělená síťové kartě, ale i tak je stále možné dostávat i legitimní data.

Zmíněný návrh tak obnáší dva hlavní body:

- Použít raw sokety
- Zajistit příjem paketů na zvolené rozhraní

Aby nová knihovna pracovala co nejrychleji, bylo nutné implementovat práci s vlákny, které umožňují paralelní zpracování dat. Pro tyto účely byla použita knihovna `libpthread`, umožňující práci s vlákny.[16] Pro zajištění druhého bodu bylo využito knihovny pro zachytávání a zpracování paketů `libpcap`, která maximálně usnadňuje práci s raw pakety.

3.2.1 Libpcap

Jedná se o knihovnu s otevřeným zdrojovým kódem, určenou k zachytávání paketů, která tvoří součást programů tcpdump, wireshark či ettercap. `Libpcap` byl vytvořen roku 1994 vědci z Univerzity v Kalifornii, kterým šlo o vytvoření platformově nezávislého API, pro možnost zachytávání paketů.[17]

Při obyčejném příjmu IP paketu jsou jádrem systému kontrolovány údaje uvnitř ethernetových a IP hlaviček a v případě, že paket dorazí do cíle, jsou hlavičky odstraněny a datová část paketu je zpracovaná aplikací. U knihoven jako `libpcap`

je paket zpracován obdobně, ale navíc je kopie paketu odeslána paketovému filtru, nacházejícímu se uvnitř jádra systému. Jedná se o jednu z implementací BPF filtrů, která umožňuje zvolit libovolný síťový port, na kterém bude naslouchánо. Je možné nastavit počet, či velikost (v bajtech) přijatých paketů. Hlavním bodem je však zmíněný BPF filtr, díky kterému je možné zachytávat pouze pakety, které projdou filtrem a není tak nutné kopírovat veškerý provoz na síťové kartě z jádra systému do uživatelské aplikace. Filtrovány jsou pak všechny příchozí pakety a zpracovány např. jen pakety s TCP příznakem nastaveným na SYN/ACK.[17, 18]

3.2.2 Libpthread

Libpthread je sada knihoven pro pracování s vláknovým API v jazyce C/C++. Je tak umožněno spustit nezávisle běžící úlohu uvnitř programu a vykonávat program paralelně k hlavnímu, což může výrazně urychlit jeho běh. Práce s vlákny je mnohem efektivnější, z hlediska režie, než spouštění nového procesu, protože sdílejí stejný adresový prostor či další datové struktury.[16]

3.2.3 Vlastnosti knihovny Tcpgen

Hlavním bodem při návrhu této knihovny bylo použít raw sokety, tzv. „SOCK_RAW“, díky kterým je možné vytvářet kompletní IP, potažmo TCP hlavičku přímo programátorem a tuto práci nepřenechávat jádru systému, který to má obvykle na starost. Je pak nutné specifikovat každý z bloků IP, TCP (příp. UDP aj.) hlaviček či dat, ale kromě spodní linkové vrstvy je tak možné mít kontrolu nad téměř celým paketem. Tím, že se IP a TCP vrstvy plní ručně a IP adresa neodpovídá použitému rozhraní, pakety vůbec nejdou dovnitř jádra (kernelu) systému a tak je nutné pro jejich zachycení naslouchat, tzv. „sniffing“, na síti a použít nějaký z BPF (BSD packet filter) filtrů, pomocí kterých by bylo možné komunikaci zachytit, viz 3.2.1. Běžně jsou totiž spojené s ovladačem, uvnitř jádra systému, pomocí kterého je možné pakety zachytit těsně před vstupem do protokolového zásobníku, kam jsou běžně směřovány. Tyto BPF filtry umožní zachytávat jen pakety, které odpovídají nastavenému filtru a není tak nutné zpracovávat všechny pakety jdoucí po síti.[17, 18] Problém nastává při použití adresy, která je přidělená na rozhranní, ze kterého je útok prováděn.

Tím, že raw sokety nenaslouchají na zvoleném zdrojovém portu, tak ani jádro systému neví, že má očekávat příjem paketů s příznaky SYN+ACK a na ty pak okamžitě odpovídá resetem spojení (paketem s příznakem RST). Tento problém by se dal odstranit s pomocí nástroje `iptables`, zablokováním všech RST paketů, jdoucích na daný server, ale ze všech hledisek je lepší vystupovat s fiktivní IP adresou a oznámit do sítě, že daná MAC adresa odpovídá ještě jiné IP adrese.

K této funkci bylo využito balíčku iputils-arping[5], který je možné nainstalovat přímo z distribučních repozitářů:

```
$ sudo apt-get install iputils-arping
```

Jedná se o nástroj umožňující zasílání ARP dotazů/odpovědí do sítě. Obsahuje však možnost odeslat i nevyžádané ARP zrávy, tzv. „Gratuitous/Unsolicited ARP“, kterými je možné podvrhnout ARP zprávy do sítě.[1] Díky těmto zprávám je tak možné oznamovat, v pravidelných intervalech, že IP adresa patří dané MAC adrese. Výchozí brána, případně cílový server, si pak uloží tento záznam do ARP tabulky a jsou tak schopni rámce směřovat na správný port.[4] Pro odeslání těchto ARP zpráv je nutné povolit následující proměnou v systému:

```
# echo 1 > /proc/sys/net/ipv4/ip_nonlocal_bind
```

Díky tomu je pak možné, aby proces volal funkci bind() s IP adresou, která není přidělená žádnému ze síťových rozhraní.

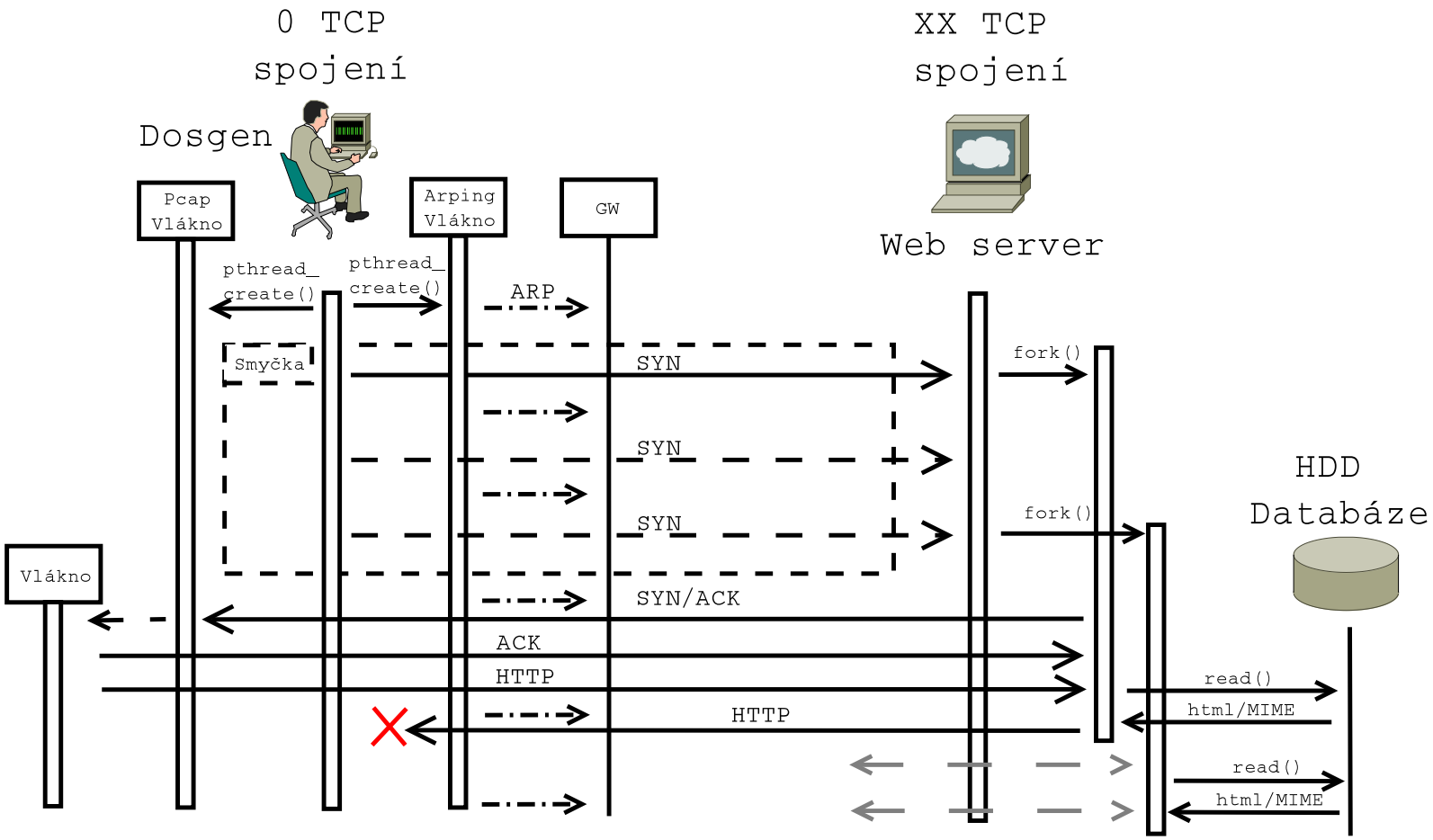
V hlavní funkci `Tcpgen` knihovny jsou zpracovány vstupní argumenty, převzaté po zpracování hlavní funkcí nástroje DoSgen, které jsou nezbytné pro správnou funkci programu. Jednou z prvních volaných funkcí je vytvoření nového vlákna, které spustí funkci `start_sniffing()`, která nastaví BPF filter, spolu se zvoleným rozhraním, pro zachytávání komunikace pouze z daného cíle útoku. Rovněž spustí funkci `pcap_loop()`, která zachycené pakety uloží do zásobníku a postupně je zpracovává.

Následně je v hlavním vlákně vytvořeno nové vlákno, funkcí `pthread_create()`, pro spuštění nástroje arping, kterému jsou předané argumenty s IP adresou zdroje, cíle a názvu použitého rozhraní. V tomto vlákně je pak v pravidelných intervalech odeslána všesměrově ARP zpráva ohlašující do sítě MAC adresu, spojenou s fiktivní IP adresou. Díky tomu jsou pak pakety směrovačem odeslány do sítě s útočícím zařízením a pomocí knihovny `libpcap` zachyceny a zpracovány.

V závislosti na zvoleném počtu spojení je v dalším kroku zavolána, ve smyčce, funkce `start_tcp_attack()`, která postupně odesílá TCP segmenty s příznakem SYN, navazující spojení. Pro každý ze segmentů je vygenerována náhodná hodnota sekvenčního čísla tak, aby byla pro každé spojení unikátní.

Další krok programu je čekání na příchozí pakety, s příznakem SYN+ACK, a jejich zpracování ve funkci `pcap_loop()`. Aby bylo zpracovávání paketů co nejrychlejší, je pro každý paket, čekající v pcap frontě vytvořeno samostatné vlákno, ve kterém je s paketem, dle typu útoku, naloženo a pak odeslán třetí segment, s příznakem ACK, čímž je spojení navázáno. Po něm už nezbývá nic jiného, než dle útoku odeslat odpovídající data. Sekvenční diagram, popisující návrh `Tcpgen` knihovny, je možné vidět na obr.3.2

[5]Ve verzi iputils-s20121221



Obr. 3.2: Návrh knihovny Tcpgen

Aby bylo možné využívat funkce knihovny `Tcpgen`, bylo nutné nějakým způsobem zakomponovat nově vytvořenou knihovnu do stávajícího modulu `LibDoS`. Při této příležitosti bylo využito nástroje *ar* a stejně jako při vytváření čistě statické knihovny byla, tato nově vzniknutá knihovna, připojena k souborům s objekty.

```
$ ar rcs -o libdos.a trafgen_wrapper.o libtcpgen.a
```

Poté už stačilo připojit, pomocí direktivy `#include`, hlavičkový soubor, obsahující deklarace funkcí z hlavního souboru `tcpgen.c` `Tcpgen` knihovny a bylo možné plně využívat funkcí z dané knihovny.
Při testování nové knihovny bylo vhodné přenastavit i maximální počet souborových deskriptorů, které je možné otevřít, na vyšší hodnotu, než byla ve výchozím nastavení.

```
$ ulimit -n
$ ulimit -n 4096
```

Aby byl dosgen v pořádku zkompilován, je případně nutné doinstalovat chybící knihovny `libpcap` a `libpthread`, pokud je již systém neobsahuje a současně s nimi i veškeré závislosti nástroje Trafgen. Dalším krokem už je kompilace nástroje DoSgen pomocí nástroje make.

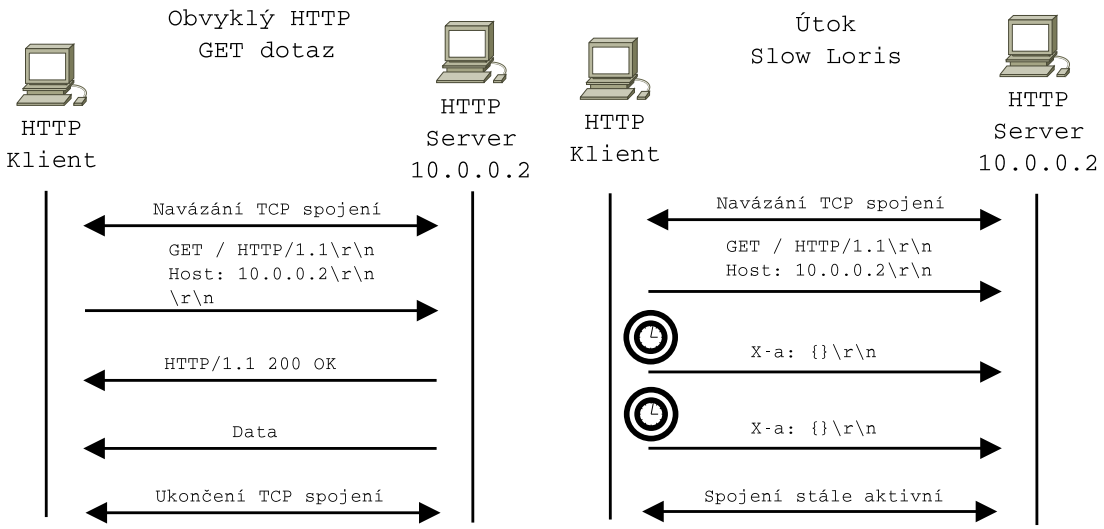
```
$ sudo apt-get install libpcap0.8 libpthread-stubs0-dev bison
    flex libnl-3-dev libnl-genl-3-dev
$ cd dosgen/
$ make
```

Implementacemi zvolených útoků byly věnovány samostatné následující sekce.

3.3 HTTP GET flood

Prvním ze zvolených útoků se stal HTTP GET flood, který jak jeho název vypovídá útočí na server pomocí protokolu HTTP. Teoretická stránka útoku je popsána v jedné z úvodních kapitol (1.4.1). Při tomto útoku je při závěrečném segmentu ACK, kterým se dokončí sestavení TCP spojení, zavolána funkce `get_flood()`. Ta se podle zvoleného počtu spojení pokusí o jeho sestavení a posléze odešle daný počet GET dotazů na určitý dokument, specifikovaný identifikátorem URI („Uniform Resource Identifier“) serveru.[4] Dotazovaný dokument je pak strategicky vybrán útočníkem, aby server co nejvíc zaměstnal a přispěl tím k DoS útoku. Útok je pak možné provést pomocí příkazu:

```
$ sudo ./dosgen -i vboxnet0 --http -s 192.168.56.150 -H
    192.168.56.101 -U /images/large_pic.jpg -C 1000
```



Obr. 3.3: Slow Loris útok

Parametry je možné libovolně měnit, avšak všechny z nich jsou povinné. Prvním z nich je `-i`, společný i pro útoky volající knihovnu `trafgen`, označuje použité rozhraní. V tomto případě se jedná o virtuální síťové rozhraní pro program Virtualbox. Druhým parametrem `--http` je vybrán útok HTTP GET flood, parametrem `-H` je specifikována IP adresa (možné i doménové jméno) cíle útoku. Dalšími parametry jsou `-U` pro specifikování URI a `-C` pro počet spojení.

3.4 Slow Loris

Dalším z útoků byl tzv. „Slow Loris“. Při něm je odeslán GET dotaz bez ukončujících CR a LF znaků na cílený server. Tím je řečeno, že dotaz není ukončen a server by tak měl očekávat příchod další části dotazu, dokud nevyprší nastavený časový limit. Následuje nastavení časového intervalu útočníka, po jehož vypršení je odesláno pokračování HTTP dotazu, tzv. „Continuation“.[8] V případě že by časovač serveru vypršel, tak by server odeslal útočníku segment s příznakem FIN, PSH, ACK, čímž by ukončil spojení. Útočníku by ale stačilo jen přenastavit hodnotu svého časovače na nějakou nižší a zkusit zaútočit znovu. Útok je vyzobrazen na obr.3.3. Výchozí hodnota časovače u apache serveru je 300 sekund.

Pro spuštění útoku jsou použity pro přehlednost stejné parametry jako v případě útoku HTTP GET flood (3.3). V tomto případě je pro změnu dotazován kořenový soubor na daném serveru.:

```
$ sudo ./dosgen -i vboxnet0 --slowloris -s 192.168.56.151 -H
    192.168.56.101 -U / -C 1000
```

3.5 Sockstress

Třetí implementovaný útok typu Sockstress není útokem na aplikační vrstvu, nýbrž dalším z útoků vyčerpávajícím serverové zdroje. Při přijímání segmentu SYN+ACK ze serveru je zavolána funkce pro odeslání posledního ACK segmentu, avšak s nastavenou „velikostí okna“ na hodnotu 0. Server je tak nucen se útočníka (nástroje DoSgen) dotazovat, kdy bude schopen přijímat nějaká data, formou opakovaných SYN+ACK segmentů. Funkce `send_ack()` je však naprogramována tak, aby posílala stále segment s „velikostí okna“ nastavenou na hodnotu 0. Útok se spouští následovně:

```
$ sudo ./dosgen -i vboxnet0 --sockstress -s 192.168.56.152 -H
    192.168.56.101 -U / -C 1000
```

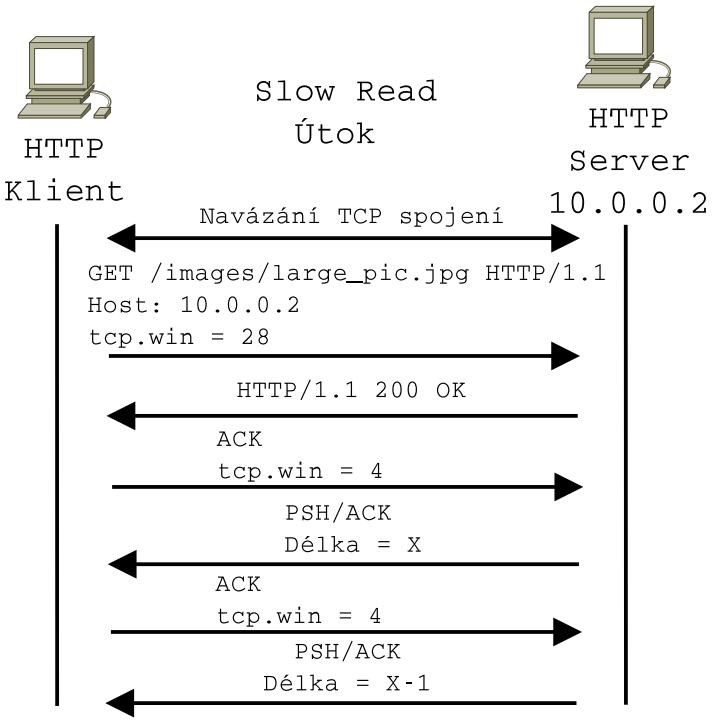
Parametry jsou obdobné jako v předchozích útocích (viz. 3.3).

3.6 Slow Read

Posledním z vybraných útoků se stal útok Slow Read. Princip je stejný jako při útoku HTTP GET flood s rozdílem, že u tohoto útoku je při odesílání HTTP dotazu (v části TCP segmentu) nastavena „velikost okna“ na velice malou hodnotu (např. 28). Jako URI je obdobně zvolen nějaký velký statický soubor, uložený na serveru, a dotaz je pak směřován na něj. Ten se však nevejde do „velikosti okna“ klienta a tak ho server nemůže odeslat nazpět v jednom kuse. Odešle jen např. HTTP hlavičku (která má velikost 28) a zbytek dat rozdělí na malé části a ty pak po jedné odesílá. Zprávy ze serveru mají nastaveny TCP příznaky na PSH a ACK, takže je jednoduché je vyfiltrovat a potvrzovat ACK zprávami funkcí `send_ack()`, ve kterých je ale opět „velikost okna“ nastavena na malou hodnotu. Takto je přenos maximálně zpomalen a server maximálně zaměstnán. Obr. 3.4 zobrazuje princip tohoto útoku.

Ve Apache verzi 2.4.7-1ubuntu4.13 server postupně odesílá zprávy s velikostí specifikované podle „velikosti okna“ oznámené útočníkem. V každém kroku pak velikost zmenší o jedničku a snaží se data odeslat. Ty však ve výsledku klientu vůbec nepřijdou.
Stejně jako v předchozích příkladech jsou při spuštění specifikovány podobné argumenty:

```
$ sudo ./dosgen -i vboxnet0 --slowread -s 192.168.56.153 -H
    192.168.56.101 -U /data/large_file -C 1000
```



Obr. 3.4: Slow Read útok

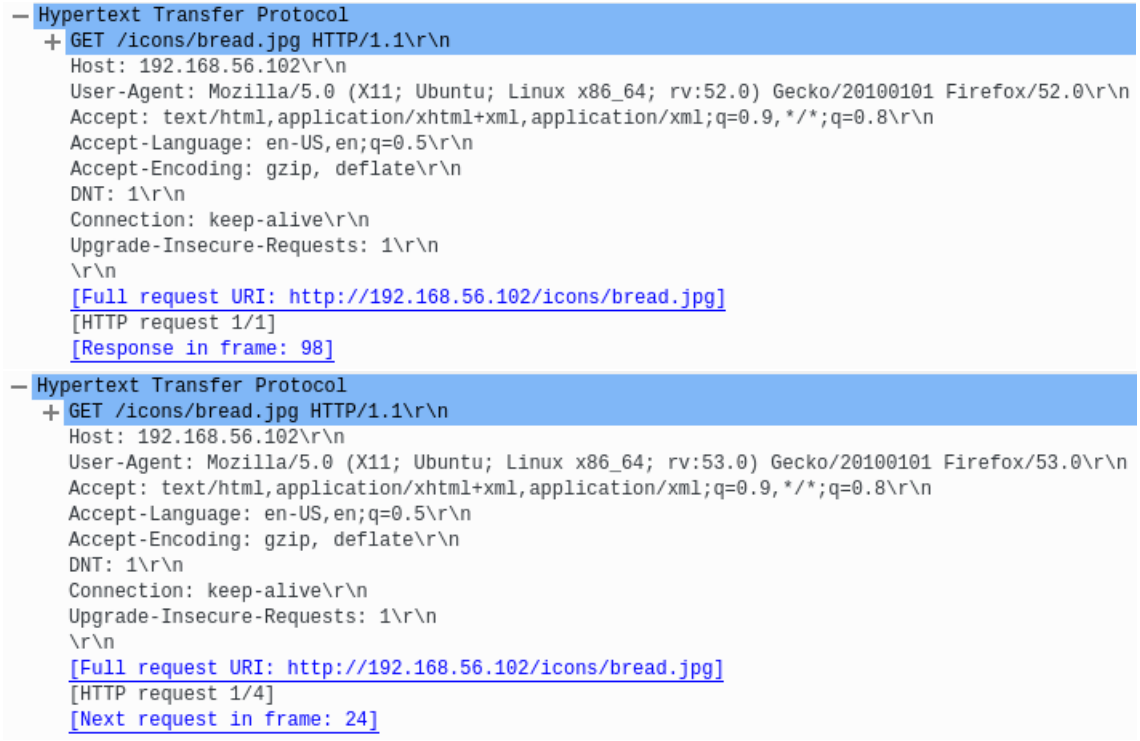
3.7 Obecné informace o Tcpgen knihovně

Nově vzniknutá `Tcpgen` knihovna využívá funkcí dvou nových knihoven, které původně nebyly součástí nástroje DoSgen. Jedná se o knihovnu pro práci s vlákny `libpthread` a o knihovnu pro možnost zachytávání paketů pomocí BPF filtru `libpcap`. `Tcpgen` knihovna využívá raw sokety, pro jejichž funkčnost je potřeba sestavit téměř celý paket, včetně části s daty. Při volání funkcí knihovny jsou očekávané vstupní parametry, předané z hlavní funkce nástroje DoSgen. Parametry jsou poté zpracovány v hlavní funkci nové knihovny a volané funkce knihovny v pořadí zahájí útok.

Tím, že jsou použity raw sokety je umožněno navázat libovolné množství spojení, protože na straně útočníka nejsou mapované zdrojové porty na IP adresu, které by byly použity při navazování spojení. Taktéž je umožněno spouštět DoSgen s libovolnou zdrojovou IP adresou, tak aby byla identita útočníka maximálně zvýšena.

Při odesílání dotazů na server pomocí protokolu HTTP byla knihovna navržena tak, aby datová část dotazů obsahovala veškeré parametry a byla tak shodná s legitimní komunikací pomocí nějakého webového prohlížeče. Srovnání HTTP dotazu při použití nástroje DoSgen a dotazu z prohlížeče Firefox lze vidět na obr.3.5. HTTP dotaz vytvořený novou knihovnou je zobrazen v horní části obrázku. Aby byl vidět rozdíl, byla v knihovně použita starší verze prohlížeče Firefox.

Je tím pádem velice obtížné filtrovat legitimní od nelegitimního provozu, z po-



Obr. 3.5: Srovnání dotazů pomocí nástroje DoSgen a prohlížeče Firefox

hledu webového serveru, pokud by byla knihovna udržována s aktuálními verzemi webových prohlížečů. V knihovně samozřejmě nechybí ani funkce pro překlad doménového jména na IP adresu, takže obsluha je maximálně usnadněna.

3.8 Testování nástroje

Nová knihovna `Tcpgen` byla otestovaná provedením útoků na dva rozdílné webové servery. Jednalo se o webový server Apache2, ve verzi 2.4.7, a Lighttpd, ve verzi 1.4.33. Oba dva servery byly otestovány jak na fyzickém, tak virtuálním pc, aby bylo zabráněno možnosti, že např. virtuální stroj bude limitován počtem spojení.

Servery byly otestovány všemi typy útoků, z nové `Tcpgen` knihovny (viz předchozí sekce), jak uvádí tab. 3.8. Protože v případě útoku HTTP GET flood závisí úspěšnost útoku na dotazovaném objektu, bylo cílem útoků při testování udržet TCP spojení aktivní po co nejdelší dobu.

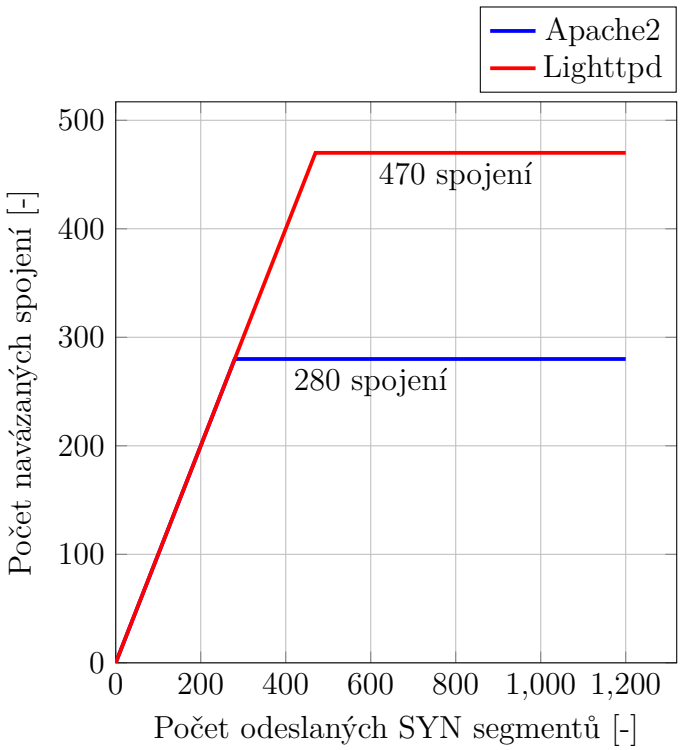
Jako příklad a grafické znázornění je zde uveden útok Slow Loris.

```
$ sudo ./dosgen -i eth0 --slowloris -s 192.168.56.150 -H
   192.168.56.102 -U / -C 800
```

U obou dvou typů serverů, při ponechání výchozí konfigurace došlo k úspěšnému DoS útoku. Jak je uvedeno v grafu 3.6, apache server byl limitován počtem 279 TCP

Tab. 3.3: Výsledky testování nástroje DoSgen z hlediska délky útoku

Typ útoku/Délka útoku [sek]	Apache	Lighttpd
Sockstress	762	69
HTTP GET flood	6	7
Slow Loris	∞	∞
Slow Read	∞	∞



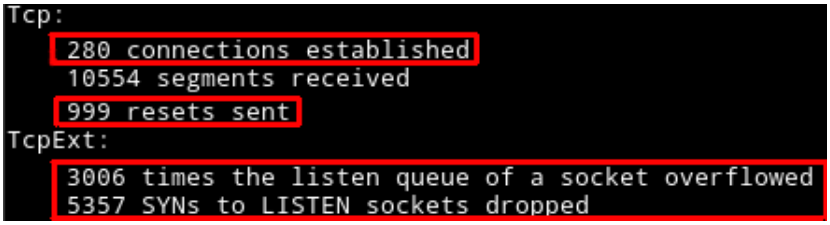
Obr. 3.6: Zátěžový test webových serverů

spojení, po jejichž vyčerpání začal nové požadavky o spojení zahazovat. Se stejným problémem se potýkal server lighttpd, který byl schopen udržet 470 aktivních TCP spojení. Statistiky TCP spojení je možné zobrazit pomocí příkazu `$ netstat -nts`. Na obr.3.7 lze vidět počet odeslaných RST spojení a např. počet SYN segmentů, které byly zahozeny.

Ani změny v systémových proměných se změnami počtu otevřených soketů, zvýšení velikosti fronty pro příchozí spojení nebo zvýšení rozsahu dostupných portů pro vytvoření soketů velkou změnu nepřinesly:

```
$ sudo sysctl net.core.somaxconn="4096";
$ sudo sysctl net.ipv4.ip_local_port_range="15000 61000"
$ sudo sysctl net.ipv4.tcp_max_syn_backlog="4096"
```

Maximální počet spojení, v případě apache serveru se zvýšil pouze na 663 a výsledkem byl opět úspešný DoS útok. U serveru lighttpd byl maximální počet souběžných spojení 2391.



Obr. 3.7: Statictiky TCP spojení

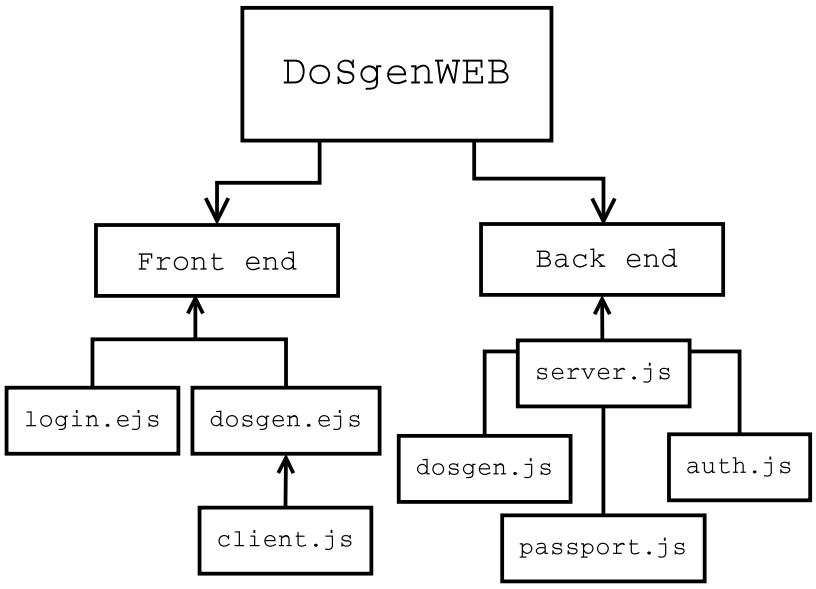
U ostatních útoků, viz tab. 3.8, záleželo na konfiguraci parametrů webových serverů a jejich časovačů. V případě útoku Sockstress došlo, po neustálém oznamování „velikosti okna“ s nulovou hodnotou, k vypršení časovače serveru. Stalo se tak po uplynutí 762 sekund (Apache), kdy server přešel do stavu FIN_WAIT1, ve kterém čekal na příchod potvrzení od klienta. Tento segment však DoSgenem nebyl odeslán, a tak server musel daný soket ponechat v čekacím stavu, než vypršela doba pro zavření soketu z důvodu vypršení časovače. Až po této době mohl server uvolnit místo pro nové spojení. Po tuto dobu byl tak server zaneprázdněn (došlo tedy k DoS útoku) a na nové požadavky o spojení neměl volnou kapacitu.

Při testování obou pomalých útoků (Slow Loris a Slow Read) bylo spojení drženo aktivní po tak dlouhou dobu, dokud nebyl nástroj DoSgen zastaven. U útoku Slow Loris začal DoSgen posílat v pravidelných intervalech (1 minuta) „Continuation“ HTTP zprávy, které byl server nucen potvrzovat segmentem ACK, dokud by nebyl HTTP dotaz ukončen znaky CR a LF. Ty však nebyly nikdy odeslány a tak server musel udržovat všechna spojení aktivní, dokud nebyly ze strany DoSgenu ukončeny. Pro útok Slow Read byla situace podobná. Po přijetí HTTP dotazu serverem, s „velikostí okna“ nastavenou na 28, server odpověděl jako při legitimní komunikaci HTTP odpovědí se statusovým kódem 200-OK (viz 1.4.1). Ten by obvykle následovaly dotazovaná data, avšak kvůli zmíněné velikosti server musel odeslat pouze HTTP odpověď bez dat a počkat na přijetí ACK segmentu ze strany DoSgenu. Oznámená „velikost okna“ DoSgenem 4B způsobila, že server nedokázal odeslat dotazovaný soubor po tak malých kusech (4 bajty) a začal data odesílat postupně s velikostí 28, kterou při každé další části zmenšoval. Díky neustálému oznamování těchto 4B způsobilo, že server začal neustále zvětšovat interval mezi datovými částmi a ve výsledku celá data nikdy neodeslal. Opět tak došlo k DoS útoku do té doby, co DoSgen potvrzoval příjem dat a ohlašoval 4B „velikost okna“.

3.9 Webové rozhraní

Dalším z bodů práce bylo rovněž rozšířit webové rozhraní nástroje DoSgen, nazvané DoSgenWEB, vytvořené pomocí systému Node.js. Jedná se o multiplatformní prostředí pro vývoj webových aplikací založeném na tzv. „V8 JavaScript engine“. Aplikace pro Node.js jsou napsané v jazyce JavaScript, které tvoří serverovou část, ale mají i tzv. „front-end“, tedy stranu klienta v šablonovém systému EJS (Embedded JavaScript).[19]

Rozhraní DoSgenWEB má taktéž část klienta a část serveru. Klientská část má na starost interakci s uživatelem a serverová část v pozadí provádí operace, jako zpracovávání argumentů nebo např. spuštění nástroje DoSgen. Na obr.3.8 lze vidět strukturu webového rozhraní.



Obr. 3.8: Struktura webového rozhraní DoSgenWEB

Hlavní soubory klientské strany jsou `login.ejs`,`dosgen.ejs` a `client.js`. Starají se o ovládací rozhraní, vytvořené v jazyce HTML, přihlašovací formulář, šablony bloků pro společné nastavení vstupních parametrů nebo pro blok útoků. Protože soubor `client.js` obsahuje definici pole s specifikacemi parametrů pro každý z útoků, rozšíření webového rozhraní se týkalo téhle části. Tento soubor se mimo jiné stará o interaktivitu aplikace, docílenou pomocí knihovny `jQuery` a technologie AJAX (Asynchronous Javascript and XML).

Strana serveru a její hlavní soubory `server.js`, `dosgen.js`, `passport.js` a `auth.js` se starají o samotný chod aplikace. Tuto součást tvoří např. spuštění samotného webového serveru, specifikace cesty k binárnímu souboru nástroje DoSgen, vypnutí procesu nebo logování výpisů.

3.9.1 Spuštění webového rozhraní

Pro spuštění samotného webového prostředí je nezbytné nainstalovat, např. z repozitářů, systém Node.js a balíčkový systém NPM (Node Package Manager).

```
$ sudo apt-get install nodejs npm
```

Teď už je zapotřebí mít v aktuálním adresáři zdrojové soubory aplikace DoSgenWEB, které obsahují soubor `package.json`, ve kterém jsou definované závislé balíčky. Následuje instalace balíčků pomocí nástroje *npm*.

```
$ sudo npm install
```

Aby mohlo být ovládací rozhraní pomocí webového prohlížeče ovládané, je nutné spustit inicializační skript `DoSgen`, který je možno uložit i do složky `/etc/init.d/`, aby byl webový server spuštěn automaticky, beheím startu počítače. [15]

```
$ update-rc.d DoSgen defaults
```

Tento skript se nachází s ostatními zdrojovými soubory v příloze práce. Rovněž je možné nalézt v příloze práce náhledy na konzolovou a webovou aplikaci (obr. C.1 a D.1).

4 ZÁVĚR

Diplomová práce se zabývala problematikou bezpečnosti v síťové komunikaci se zaměřením na DoS/DDoS útoky. Hlavním z cílů bylo vytvoření softwaru v jazyce C, který bude schopen generovat vybrané DoS útoky v oblasti od transportní po aplikační vrstvu ISO/OSI modelu. Byly proto vybrány útoky Sockstress, HTTP GET flood, Slow Loris a Slow Read.

Úvodní kapitola se věnovala vysvětlení problematiky DoS/DDoS a zejména pak teoretickému popisu útoků, od těch běžných, zejména na nižší vrstvy ISO/OSI modelu, po složitější útoky na aplikační vrstvě, cílené na některý nedostatek aplikačního protokolu. Ke každému z útoků byla uvedena i slabina, které daný útok zneužívá.

Druhá část práce se věnovala volně dostupným nástrojům pro uskutečnění DoS útoků. Každý z nástrojů byl popsán z teoretického pohledu a byly zde sepsány i kroky, jak je možné jej získat a případně pak použít. Jsou to nástroje packETH, Trafgen a Nping. Nástroje byly v další části porovnávány se zaměřením na počet vygenerovaných paketů za sekundu. Pro srovnání bylo měřeno jak pro Gigabit Ethernet, tak pro Fast Ethernet. Konkrétní výsledky jsou uvedeny na konci kapitoly, nejlepšího výsledku však dosáhl nástroj Trafgen.

Třetí kapitola se zabývala rozšířením stávajícího nástroje DoSgen, napsaném v jazyce C, který jako své jádro používá právě nástroj Trafgen. Byl zde vysvětlen jeho konfigurační jazyk a který z jeho modulů se stará o vytváření paketů. Tím, že nástroj Trafgen nebyl původně koncipován pro legitimní komunikaci a výměnu zpráv se serverem, bylo nutné DoSgen rozšířit o nový modul, který by nějakým způsobem implementoval sestavení TCP spojení. Byla proto vytvořena nová knihovna Tcpgen, která rovněž obsahuje implementaci zmíněných útoků, jejichž součástí je sestavení platného paketu s daty, včetně ošetření správných vstupních hodnot programy. Nová knihovna byla navržena tak, aby bylo možné vystupovat s fiktivní IP adresou a komunikace byla velice těžko rozpoznatelná od legitimní. Nově rozšířený nástroj byl otestován útoky na dva různé webové servery a v obou případech vyčerpal dostupný limit TCP spojení, i v případě zvýšení systémových limitů.

Závěrečná část byla věnována rozšíření ovládacího rozhraní nástroje DoSgen a popisem jeho struktury. Rovněž bylo popsáno ovládání a instalace tohoto rozhraní, včetně všech závislostí.

LITERATURA

[1] PROWELL, Stacy J., et al. Seven deadliest network attacks. Vyd. 1. Boston: Syngress, 2010. 176s. ISBN 15-974-9549-2.

[2] Types of DDoS Attacks. Verisign iDefense Threats [online]. 2015, [cit 27.11.2016]. Dostupné z URL: <https://www.verisign.com/en_US/security-services/ddos-protection/types-of-ddos-attacks/index.xhtml>.

[3] KENIG, R., DDoS Survival Handbook. Radware [online]. 2013, [cit. 28. 11. 2016]. 56 s. Dostupné z URL: <http://security.radware.com/uploadedFiles/Resources_and_Content/DDoS-Handbook/DDoS_Handbook.pdf>.

[4] BURDA, Karel. Návrh, správa a bezpečnost počítačových sítí. Brno, 2014. ISBN 978-80-214-5155-1.

[5] KOZIEROK, Charles M. *The TCP/IP Guide: A Comprehensive, Illustrated Internet Protocols Reference*. Vyd. 1. San Francisco: No Starch Press, 2005, 1616 s. ISBN 159327047X.

[6] HUTCHENS Justin: Kali Linux Network Scanning Cookbook. Vyd. 1. Birmingham: Packt Publishing Ltd., 2014. 400s. ISBN 978-1-78398-214-1.

[7] FIELDING, R. , et al. *Hypertext Transfer Protocol – HTTP/1.1*. RFC 2616, Červen 1999. Dostupné z URL: <http://www.rfc-editor.org/rfc/rfc2616.txt>

[8] Jelic Filip: Single Computer DoS – Slow Loris Attack. [online]. 2017, [cit 5.5.2017]. Dostupné z URL:<https://www.deepdotweb.com/2017/01/19/single-computer-dos-slow-loris-attack/>

[9] Sergey Shekyan: Are you ready for slow reading? [online]. 2012, [cit 20.4.2017]. Dostupné z URL: <https://blog.qualys.com/securitylabs/2012/01/05/slow-read>

[10] PACKETH. [online]. [cit. 6.12.2016]. Dostupné z URL: <http://packeth.sourceforge.net/packeth/Home.html>.

[11] netsniff-ng toolkit. [online]. [cit. 20. 10. 2016]. Dostupné z URL: <http://netsniff-ng.org/>.

[12] SCHIPP, J. A Look at the Netsniff-NG Toolkit, A Suite of High-Performance Networking Tools. [online]. [cit. 23.1.2016]. Dostupné z URL: <http://www.draconyx.net/talks/mosscon2013.pdf>

[13] Nping. Introduction. [online]. [cit. 4.12.2016]. Dostupné z URL: <https://nmap.org/>

[14] The Trafgen Expression Language. Draconyx, LLC [online]. 2015, [cit. 17.11.2016]. Dostupné z URL: <http://www.draconyx.net/articles/trafgen-expression-language.html>.

[15] Halaška, P.: Generátor kybernetických útoků. [s.l.], 2016. 58s. Vedoucí diplomové práce Ing. Jan Hajný, Ph.D. Dostupné z URL: <https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id-125791>

[16] POSIX thread (pthread) libraries. [online]. [cit 3.4.2017]. Dostupné z URL:<http://www.yolinux.com/TUTORIALS/LinuxTutorialPosixThreads.html

[17] Luis Martin Garcia: Programming with Libpcap - Sniffing the Network From Our Own Application. [online]. 2008, [cit 2.5.2017]. Dostupné z URL: <http://recursos.aldabaknocking.com/libpcapHakin9LuisMartinGarcia.pdf>

[18] ERICKSON, Jon. Hacking the art of exploitation. Vyd. 2. San Francisco: No Starch Press, 2008. 488s. ISBN 978-1-59327-144-2.

[19] CANTELON, M.: et al. Node.js in Action. Shelter Island: Manning Publications Co., 2014. 417s. ISBN 9781617290572.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

DOS Denial Of Service

DDOS Distributed Denial Of Service

ISO/OSI International Organization For Standardization/Open Systems Interconnection

HTTP Hypertext Transfer Protocol

TCP/IP Transmission Control Protocol and Internet Protocol

UDP User Datagram Protocol

ICMP Internet Control Message Protocol

SYN Synchronization flag

SN Sequence Number

AN Acknowledgment Number

MITM Man In The Middle

ACK Acknowledgment flag

RST Reset flag

PSH Push flag

FIN Finish flag

HTTPS Hypertext Transfer Protocol Secure

DNS Domain Name System

SMTP Simple Mail Transfer Protocol

HTML Hypertext Markup Language

CPU Central Processing Unit

RAM Random Access Memory

LTS Long Term Support

MAC Media Access Control

ARP Address Resolution Protocol

BPF BSD Packet Filter

PCAP Packet Capture

API Application Program Interface

RST Reset

CR Carriege Return

LF Line Feed

URI Uniform Resource Identifier

WIN Window

AJAX Asynchronous Javascript and XML

XML Extensible Markup Language

NPM Node Package Manager

SEZNAM PŘÍLOH

A ARGUMENTY NÁSTROJE DOSGEN

Tab. A.1: Možné typy útoků pro Trafgen/Tcpgen jádro

Argument	Popis
`--syn`	SYN flood
`--rst`	RST flood
`--udp`	UDP flood
`--icmp`	ICMP flood
`--arp`	ARP flood
`--dns`	DNS flood
`--dhcp`	DHCP starvation

Argument	Popis
`--http`	HTTP GET flood
`--sockstress`	Sockstress
`--slowloris`	Slow Loris
`--slowread`	Slow Read

Tab. A.2: Možné argumenty nástroje DoSgen pro Trafgen/Tcpgen jádro

Argument	Popis
`-i`	Použité rozhraní
`-P`	Počet jader
`-s`	Zdrojová IP adresa
`-d`	Cílová IP adresa
`-S`	Zdrojový port
`-D`	Cílový port
`-n`	DNS jméno
`-p`	Výplň
`-h`	Pomocný výpis

Argument	Popis
`-i`	Použité rozhraní
`-s`	Zdrojová IP adresa
`-H`	Cílová IP adresa/Doménové jméno
`-U`	URI dotazu
`-C`	Počet spojení

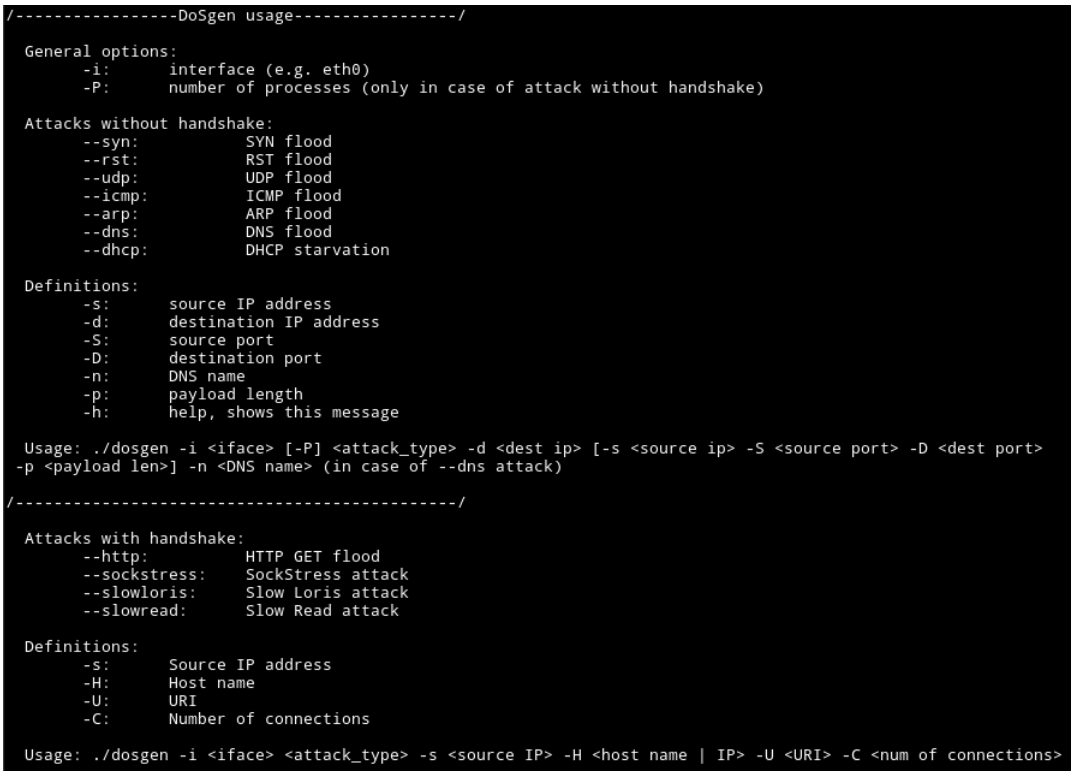
B OBSAH PŘILOŽENÉHO DVD

Struktura DVD:

- zdrojové soubory nástroje DoSgen
  - `dosgen.c`
  - `libdos.h`
  - `Makefile`
  - `README.md`
  - `trafgen_configs.h`
  - `trafgen_wrapper.c`
  - `trafgen_wrapper.h`
  - `trafgenlib.h`
- zdrojové soubory nástroje Trafgen
- zdrojové soubory nástroje Tcpgen
  - `checksum.c`
  - `checksum.h`
  - `handshake.c`
  - `handshake.h`
  - `tcpgen.c`
  - `tcpgen.h`
- inicializační skript webového rozhraní
- zdrojové soubory pro webové rozhraní
  - `bin/`
  - `cert.pem`
  - `config.json`
  - `key.pem`
  - `node_modules/`
  - `package.json`
  - `passport.js`
  - `public/`
  - `routes/`
  - `server.js`
  - `views/`
- elektronická verze práce ve formátu PDF
- virtuální disk pro virtualizované prostředí VirtualBox

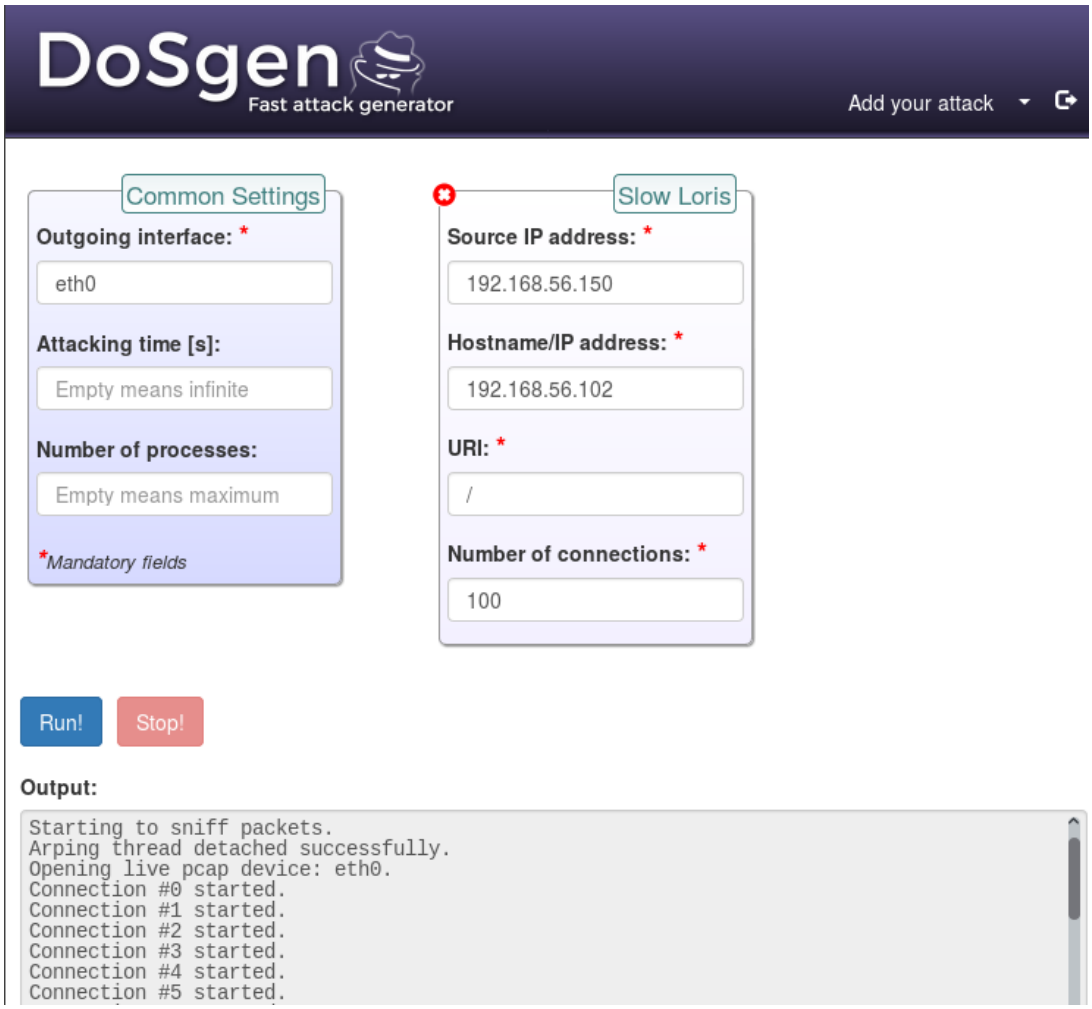
Přihlašovací údaje do virtuálního stroje jsou user:user. Pro demonstraci je po spuštění služby DoSgen (Inicializační skript) možné přihlásit se k webové aplikaci na tomto stroji pomocí webového prohlížeče a URL: „`https://$IP_ADRESA:9999/dosgen`“. Údaje pro přihlášení jsou stejné jako pro příhlášení do virtuálního stroje (user:user).

C NÁHLED NA NÁSTROJ DOSGEN



Obr. C.1: Pomocný výpis pro nástroj DoSgen

D NÁHLED NA WEBOVÉ ROZHRANÍ NÁSTROJE DOSGEN



Obr. D.1: Webové rozhraní aplikace DoSgenWEB